Defense Mechanisms for Large Language Model Security Vulnerabilities: A Literature Review

Mohankumar Muthusamy
School Of Computing
Dublin City University
Dublin, Ireland
mohankumar.muthusamy2@mail.dcu.ie

***Abstract*: Large Language Models (LLMs) are being increasingly integrated into daily use applications in a variety of fields, making security and robustness concerns important. Because LLMs interact with humans through natural language, LLMs are susceptible to attacks that target the model via a natural language input prompt. The literature review of defenses against unsafe LLM interactions is organized into four levels of defenses: input-level defenses (to prevent dangerous prompts), model-level defenses, output-level defenses, and system-level defenses (to prevent dangerous LLM outputs). The review compares various defense metrics such as Attack Success Rate (ASR) and False Positive Rate (FPR). Despite the improvements, adaptive multi-turn attacks, which adapt during the course of a conversation, are a continuing challenge for researchers. This suggests that context-aware defensive systems with adaptation to the evolving behavior of malicious actors in real time are necessary. Reinforcement-driven and memory-aware approaches could be key to keeping LLMs secure over time in changing conditions.**

Keywords— Large Language Models, Security, Defensive Mechanisms, Robustness, Adversarial Defense

I. INTRODUCTION

In today's modern world, LLMs have become integral to daily decision-making. Nowadays, LLM is being used in all professions such as doctors, lawyers, teachers, software engineers and even general public use LLM in their day-to-day life to make a decision. Also, lot of organizations has started to use LLM in their field and merge it with their existing and new products to achieve their organizational goals. So, the developers play a key role in developing a secured LLM and make sures these applications can be used by general people and ensures the security features are safe and sound because they don't know that the information given by LLM is not reliable and they take these decisions blindly suggested by LLM. Since many users are lack in technical understanding especially school kids which leads to unsafe or misguiding outputs can cost a serious consequence. Similarly, users unintentionally exploit LLM to generate harmful or unethical content. Therefore, the security precautions in LLM should function in a way where it should protect its users from unsafe contents and safeguard the model from these issues. Usually, these conversations between users and LLM's primarily occurs through natural language and securing their conversations will form the foundation of LLM security.

This review aims to evaluate existing LLM defense mechanisms and identify gaps for future adaptive frameworks.

A. Importace of Security in LLMs

Security in Large Language Models (LLM's) is way more important than in securities like data privacy, digital identity thefts or in financial sectors because LLM gives people to access to every aspect of knowledge. So, if there is any security weakness can cause serious problems and consequences which cannot be handled by the traditional security methods. As many organisations and countries around the globe are competing to develop and introduce AI systems. So, there will be a lot of security flaws and also testing these LLM's is also a difficult task because it's hard to predict how real users will have conversations with these LLM.

II. SECURITY LANDSCAPE OF LARGE LANGUAGE MODELS

A. Overview of LLM Vulnerabilites

Large language models are vulnerable to different types of attacks. They are categorised into two major categories, data level and usage level[1]. It is categorised based where and when the attack is happening.

Data and Model level:

Sensitive data leakage is a major problem, the model mistakenly reveals the actual data it trained on when the user prompt the model[1].

Model inversion is also similar to the previous one , where it tries to reconstruct the dataset from the response of the Model[1].

Model extraction , using the repeated prompt and getting the response and it analyse it to reconstruct the cloned model[1].

Data poisoning is an process where the attacker infiltrate the system and inject the malicious data into the training dataset, it leads model learned from incorrect data, finally it leads to the misleading response from the model when user queries it[1].

Backdoor attacks , similar to the data poisoning , the attacker injects the malicious trojan keyword into the dataset, where it remains inactive in the system until it gets triggered by the specific sequence of words or characters, once it is triggered the model starts behaving unexpectedly[1].

Usage and interaction level:

Prompt injection , manipulates the model's reasoning by introducing cleverly crafted tricky prompts that override or bypass the internal safety instructions of an LLM[1].

Membership inference attacks attempt to deduce whether specific private data instances were part of the training data, lead to PII data leaking from training data, usually because of improper cleaning and masking of training data[1].

Reinforcement learning–based (RL) attacks adapt dynamically by learning from previous model responses, gradually understanding how the model behaves and it try the bypass the rules and safety instructions[1].

Jailbreak attacks override system-imposed restrictions through techniques such as role-playing or "Do Anything Now (DAN)" instructions [1][6].

A recent evolution of this approach, termed deceptive jailbreak attacks, hides malicious intent inside the harmless prompts, tricking the LLM into producing restricted or unsafe outputs considering it was a harmless prompt [2].

B. Need for Defensive Measures

Continuous rapid evolution of attack techniques nullifies the existing defensive strategy. It is an ongoing security issue for LLMs. Though the traditional safety mechanisms, like input filtering and input pre-processing, and other model fine-tuning for these attacks are in practice, their efficacy is limited to the existing adversarial attacks[3][4]. Recent studies highlight the rise of attack methods like adaptive multi-turn attacks [2] , where they used an In Context Learning (ICL) method with LLM to generate the attack prompts to simulate the attack scenarios. It is a feedback-based reinforcement learning where it makes the attack prompt adaptively based on the LLM's previous response. Similarly, they defended the LLM via the same feedback mechanism using a judge LLM. If the attack LLM have more memory context than the Judge LLM, the actual LLM is vulnerable to the attack LLM. This demonstrates the need for an advanced defensive strategy which need be active in a given memory context.

III. Defesne Mechanisms For Large Language Models

Securing Large Language Models (LLMs) involves multiple layers of defense that work together to protect the model and the user. The main categories include input-level, model-level, output-level, and system-level defenses.

Each layer focuses on a specific point of protection — from filtering unsafe prompts to maintaining security at deployment.

A. Input Level Defenses

Input-level defenses are the first layer of an security. It focus on analysing the prompt input before it sent to the LLM.It includes input preprocessing and input filtering. It prevents the Usage and Interaction levels attack such as prompt injections and jailbreak attacks. In the work Improving LLM Outputs Against Jailbreak Attacks with Expert Model Integration [6], they use the small expert model called Archias, which validates the each user prompt and classify it as in-domain, out-of-domain or malicious. The result of classification is embedded into the prompt before sending it to the model guiding the guiding model to give the better possible response. This type of mechanisms helps to pro actively eliminates the threats to an LLM.

B. Model-Level Defenses

Model-level defenses focus in increasing the model efficacy against these attacks. It is an strengthening of the models inbuilt defense mechanisms. It is done via fine tuning or training the model against these attacks and training the model how to handle these malicious prompt. In Continual Defense Against Evolving Jailbreaks: A Multi-Agent Adversarial Framework with Linear Gating MoE [2], the authors propose a Mixture-of-Experts (MoE) design, where the model's is divided into smaller experts groups, known as expert agent. It activates the specific expert agent to handle the specific type of attack. It also uses the In Context Learning(ICL), to make the model not to forget the old attacks. Llama Guard, an fine-tuned model developed by Meta specifically to handle malicious prompts.

Model-level defenses form the foundation for long-term protection by making the model independently protect itself without relying on external defense services.

C. Output Level Defenses

Output level defenses usually work after a model has already generated its response. The primary goal of output level defense is to monitor and filter the final output that LLM provides and to make sure it does not contain any unsafe or unethical information before providing it to the user. The paper "Auto Defense: Multi-Agent LLM defense against Jailbreak Attacks" [5] explains that a system uses several smaller models that work together to check the main model's response. These smaller models review what the main model response, and if they find anything that violates any rules or providing any harmful content, that part of the response will be either modified or their request gets rejected. The experiments in this paper shown that this method helped lower GPT-3.5's jailbreak attack success rate from 55.74% to 7.95%, which shows a good improvement in the safety measures. These types of defenses are generally measured using False Positive Rate (FPR) and Attack Success Rate (ASR) reduction. Both FPR and ASR help to keep a good balance between the model's safety and natural quality of its language.

D. System Level Defenses

System level Defences focuses on the security aspects that go beyond the model itself. They actually deal with how Large Language Models (LLM's) are implemented, managed and protected in real world environments. These defences look into external factors like user access control, how it is monitored, and whether it follows standard cybersecurity frameworks. The paper "Vulnerability Detection in Large Language Models: Addressing Security Concerns" [3] states that LLM security issues are using well known standards like OWASP, MITRE ATLAS and NIST. The study points out specific risks such as LLM03 Supply Chain Risk and LLM06 Excessive Agency. These two risks arise when it depends on third-party services or giving the model too much control over automated processes, which will create security weakness.

System Level Defences use various methods to protect the system, such as authentication (verifying user identity), audit logging (keeping track of activities), red teaming (testing the system for weaknesses), and frequent monitoring (watching for any type of threats in real-time). These steps help to keep the system secure during every stage of implementation.

Even if model works safely, these defences will make sure that the operating environment remains trustworthy. This layer connects all parts of the defence systems and gives overall structure and supervision that supports technical defences at other levels.

E. Summary

Input-level defenses prevent harmful prompts from reaching the model, model-level techniques build internal strength, output-level systems filter unsafe responses, and system-level mechanisms manage deployment and governance. Each layer works toward a common goal ensuring safe, reliable, and responsible use of large language models.

IV. Comparative Analysis

Each defense strategies for LLMs different in where and how they are enforced. The efficacy, role, performance varies for each. Input level defenses like [6], prevent the prompt injection by flagging the harmful content before sending to the model. Their efficacy directly depends on the how the harmful is content is flagged. Incorrect flagging leads to the increase in False Positive Rate. Model level defense like [2] focus on strengthening the model capability to handle the harmful prompts through fine tuning. The newer needs a model retraining which requires a considerable amount of time and computing resources. Output level defenses like Autodefense [5], uses judge model to validate the output before sending to the user. However, the additional model validation increases the overall performance and response time in high scale. System level defenses like [3], securing the LLM infrastructure by following the governance and operational safety standards provided by OWASP, MITRE and NIST. Finally, the overall comparative analysis suggests there is no given approach to handle the adaptive attacks like multi turn attack with the given memory context.

V. Research Gaps And Future Directions

Although significant progress has been made in building defense mechanisms for Large Language Models (LLMs), there is some research gaps remain, specifically against the evolving threats like Adaptive Multi-Turn Attacks (AMTAs). The traditional defense mechanisms like input filtering, input preprocessing and fine tuning against the harmful prompt will defend only against the existing known attacks. But the AMTAs continuously learn from the model's previous response and understand the model behavior and its weakness, using reinforcement-style feedback and In-Context Learning (ICL) to bypass conventional defense layers. From the study [5], if the attacker model has larger memory context than the victim model memory context, the victim model cannot hold the defense context history, which will make it vulnerable to AMTA attack.

The Future research should be focusing on the Adaptive Defense Frameworks . Memory-aware defense will learn and analyze the intent of the user from the conversation continuously and give the real time intent risk scoring. Addressing these gaps will enhance the security of an LLM in the dynamic and complex environment.

VI. Conclusion

The security of Large Language Models (LLMs) continues to be a critical research area as their adoption expands across real-world applications. This paper reviewed key defense mechanisms across input, model, output, and system levels, highlighting how each contributes to reducing risks from adversarial and prompt-based attacks. While these defenses have improved safety and robustness, adaptive multi-turn attacks remain a major challenge. These attacks exploit conversational memory and feedback, allowing them to evolve beyond traditional one-turn defenses. To address this, future defensive strategies must become adaptive, context-aware, and capable of learning dynamically through continuous interaction. Building such memory-sensitive and reinforcement-driven defense models will be essential to sustain the trust, reliability, and safe deployment of LLMs in complex environments.

References

[1] F. Liu, J. Jiang, Y. Lu, Z. Huang, and J. Jiang, "The ethical security of large language models: A systematic review," *Frontiers of Engineering Management*, vol. 12, no. 1, pp. 128–140, Mar. 2025, doi: 10.1007/s42524-025-4082-6.

[2] Q. Xu and F. Liu, "Continual Defense Against Evolving Jailbreaks: A Multi-Agent Adversarial Framework with Linear Gating MoE," in *2025 5th International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*, June 2025, pp. 1409–1413. doi: 10.1109/CAIBDA65784.2025.11182780.

[3] S. Ben Yaala and R. Bouallegue, "Vulnerability Detection in Large Language Models: Addressing Security Concerns," *Journal of Cybersecurity and Privacy*, vol. 5, no. 3, 2025, doi: 10.3390/jcp5030071.

[4] H. Lin, Y. Lao, T. Geng, T. Yu, and W. Zhao, "UniGuardian: A Unified Defense for Detecting Prompt Injection, Backdoor Attacks and Adversarial Attacks in Large Language Models." 2025. [Online]. Available: https://arxiv.org/abs/2502.13141

[5] Y. Zeng, Y. Wu, X. Zhang, H. Wang, and Q. Wu, "AutoDefense: Multi-Agent LLM Defense against Jailbreak Attacks." 2024. [Online]. Available: https://arxiv.org/abs/2403.04783

[6] T. Tsmindashvili *et al.*, "Improving LLM Outputs Against Jailbreak Attacks With Expert Model Integration," *IEEE Access*, vol. 13, pp. 134976–134988, 2025, doi: 10.1109/ACCESS.2025.3592458.